

Bill Griffee

NETWORK ATTACHED STORAGE

Bachelor's Thesis
Information Technology


May 2013



MIKKELIN AMMATTIKORKEAKOULU

Mikkeli University of Applied Sciences

DESCRIPTION

 <p>MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences</p>		Date of the bachelor's thesis 13.5.2013	
Author(s) Bill Griffiee		Degree programme and option Information Technology	
Name of the bachelor's thesis Network Attached Storage			
Abstract <p>Digital distribution of media has grown increasingly important as an additional requirement is to have this media available to multiple users. Traditionally, servers have filled this role in companies by providing storage that can be accessed by other devices on a network, typically through file sharing or serving or via a storage area network, while providing other services as well. Network attached storage (NAS) systems expand upon traditional servers in that they are built specifically with file sharing as their primary role and are usually limited to just that. The benefit of this is that by being built specifically for this purpose, NAS systems can fulfill these needs with less powerful hardware than what was necessary for a full server.</p> <p>This thesis will be looking into various forms of NAS systems, the protocols used on them and how they are commonly used. A comparison will be made of several software-based NAS solutions and the one most suited to the customer's requirements will be implemented. The requirements of the customer, Mikkeli University of Applied Sciences, are a NAS system that can provide shared storage via SMB/CIFS and NFS, act as an iSCSI target and must utilize a PCI-express based hybrid hard drive which uses solid state flash memory as a cache for the standard hard drive attached to it.</p>			
Subject headings, (keywords) network, storage, NAS, smb, cifs, nfs, samba			
Pages 34	Language English	URN	
Remarks, notes on appendices			
Tutor Matti Jutilainen		Employer of the bachelor's thesis Mikkeli University of Applied Sciences	

CONTENTS

1	INTRODUCTION	4
2	NETWORK ATTACHED STORAGE	4
2.1	Standard computer based systems	6
2.2	Embedded systems	7
2.3	Communication and Network Protocols	8
2.3.1	Server Message Block/Common Internet File System (SMB/CIFS)	9
2.3.2	Samba	10
2.3.3	Network File System (NFS)	11
2.3.4	Apple Filing Protocol (AFP)	11
2.3.5	File Transfer Protocol (FTP)	12
2.3.6	Web-based Distributed Authoring and Versioning (WebDAV)	12
2.4	Direct Attached Storage and Storage Area Networks	13
2.4.1	Direct Attached Storage (DAS)	13
2.4.2	Storage Area Network (SAN)	13
2.4.3	SAN Protocols and Architectures	14
2.5	Wake-On-LAN (WOL)	17
3	NAS OPERATING SYSTEMS	17
3.1	Linux, FreeBSD, illumos and Other “Unix-like” Operating Systems	18
3.1.1	OpenFiler	19
3.1.2	OpenMediaVault	19
3.1.3	Linux + iSCSI and Other Additions	20
3.1.4	NexentaStor	20
3.1.5	FreeNAS and NAS4Free	20
3.2	Windows Based Operating Systems	21
3.3	OS X Server	21
3.4	Operating System Comparison	21
4	INSTALLING, TESTING AND FINAL IMPLEMENTATION	24
4.1	Installation Details	24
4.2	Testing and Comparisons	26
4.3	Final Implementation	29
5	CONCLUSION	32

BIBLIOGRAPHY	35
APPENDIX	39
1 Abbreviations	39

1 INTRODUCTION

Digital storage needs are greatly increasing worldwide as more and more digital media is created. Many companies are now required by their nation's laws to keep digital records as well as hospitals and government agencies. Professional and casual photographers have moved to digital photography in large numbers and digital videos are now standard as well. Digital distribution of media has grown increasingly important as an additional requirement is to have this media available to multiple users. Traditionally, servers have filled this role in companies by providing storage that can be accessed by other devices on a network, typically through file sharing or serving or via a storage area network, while providing other services as well. Network attached storage (NAS) systems expand upon traditional servers in that they are built specifically with file sharing as their primary role and are usually limited to just that. The benefit of this is that by being built specifically for this purpose, NAS systems can fulfill these needs with less powerful hardware than what was necessary for a full server.

This thesis will be looking into various forms of NAS systems, the protocols used on them and how they are commonly used. A comparison will be made of several software-based NAS solutions and the one most suited to the customer's requirements will be implemented. The requirements of the customer, Mikkeli University of Applied Sciences, are a NAS system that can provide shared storage via SMB/CIFS and NFS, act as an iSCSI storage target and must utilize a PCI-express based hybrid hard drive which uses solid state flash memory as a cache for the standard hard drive attached to it.

This thesis begins in chapter 2 by defining what network attached storage is, what features are common to the systems, the protocols most important to these systems and other features that may be of importance to those deploying NAS systems. Chapter 3 covers in detail the various operating systems used for NAS systems, particularly those that are user-installable. It then goes further into detailing various issues encountered with them during the writing of this thesis as well as some as comparisons. In chapter 4, the final implementation of NAS server for the client will be detailed and finally, chapter 5 will cover the conclusion of this thesis.

2 NETWORK ATTACHED STORAGE

The definition of a network attached storage (NAS) device varies some from source to source. Most agree upon the following: "NAS systems are specialized file servers that are set up, built

or designed specifically for the purpose of sharing files over a network.” Commercial NAS systems typically come with preconfigured hardware and operating systems while non-commercial alternatives are more of a “do-it-yourself” variety that use heavily modified or preconfigured operating systems sometimes known as “NAS software”. NAS systems support multiple file-based protocols and all are capable of being remotely administered over a network, some of them exclusively so, in particular embedded systems (Lehmann 2007, 91; Waring 2007; Shelly & Vermaat 2012, 749.)

NAS systems come in two essential forms:

- Standard computer based systems
- Embedded systems

Both types have their advantages and disadvantages, usually related to cost, power usage, expandability, size and speed although these distinctions are not always clear in some cases due to the large variety of NAS systems available. Regardless of their differences, all NAS systems have, at the very least, a CPU (central processing unit), some form of storage to be shared on a network, a storage location for the operating system, RAM (random access memory) for the operating system to run from and a network interface for communication. RAID (Redundant Array of Independent Disks) is another feature commonly available to both types that works by basically combining or pooling multiple storage disks in various ways so that they appear as one to the operating system (Carpenter 2011, 64). NAS systems are sometimes known as “storage appliances” as they are meant to perform the single task of providing file storage over a network and can expand upon that storage simply by connecting in more NAS devices (Shelly & Vermaat 2012, 749).

One term that sometimes comes up in documentation about NAS systems is “NAS head”. Multiple definitions exist for this as well. Lehman (2007, 93) defines it as a “...NAS which does not have any on-board storage, but instead connects to a SAN.” and that it acts as translator between file-level protocols and block-level protocols, both of which are covered in more detail in chapters 2.3 and 2.4. Heger (2008, 58) defines it differently as “...the part of the NAS solution required for clients to connect to the IO (input/output) subsystem.” or basically the NAS device itself not including the actual physical storage. The latter definition seems to be more common though and is also stated as being a “NAS Gateway” in the same document.

One interesting development in NAS design is the usage of application-specific integrated circuits (ASICs) to implement many of the functions more commonly handled by the CPU. In an article by O’Keefe (2012), it describes how Hitachi’s NAS systems achieve higher throughput by offloading tasks as networking, file system and storage operations to field-programmable gate arrays (FPGAs) which are basically programmable ASICs. Another benefit is that the FPGAs operate on their own communication buses that are separate from others in the system preventing reductions in performance that can occur on a shared bus as would happen in a normal computer. The article further details how pipelining is used between the FPGAs to further increase performance by allowing “...many operations to proceed in parallel across multiple, independent memory banks and FPGA chips, greatly increasing performance, stability under heavy load and power efficiency.” This offloading of tasks is similar to that of other applications using ASICs such TCP offload engines in network interface cards which move much of the processing of TCP traffic away from the CPU (Crowley et al. 2005, 81) only on a larger scale.

2.1 Standard computer based systems

Standard computer based NAS systems are usually the fastest but are also often the most expensive. These are usually x86/x64 processor based systems that are either available as preconfigured devices from many manufacturers (IBM, Cisco, Synology, QNAP, LaCie to name a few) or can be constructed from specialized or standard computer parts. Because they are standard computer based, they typically have more memory, higher performance hardware and more capability for expansion. Adversely, they typically consume more power although systems designed with low power consumption CPUs such as Intel’s ATOM, AMD’s Fusion and VIA’s Nano processors can compete favorably in this regard with embedded systems using ARM and other non-x86 processors (Dang & Angelini 2012; Stokes 2010.)

The operating system running on these systems vary greatly. Modified versions of Linux and FreeBSD are common (e.g. Openfiler, FreeNAS), Microsoft offers Windows Storage Server as a NAS operating system, and some companies use other versions of Windows Server (IBM, Dell). Some companies also develop their own proprietary operating systems, some of which are based on Linux or FreeBSD (e.g. QNAP). Several of these operating systems will be covered in more detail later as one of these will be selected for use in the final implementation of a NAS system.

A wide variety of designs exist of these as any standard computer can be set up as one. Rack mounted servers, standard personal computers (PCs), commercial proprietary devices, and small devices using mini-ITX form factor boards all exist and vary greatly in price and capability. Configuring a standard PC as a NAS system will be covered in more detail later in chapters 3.5 and 3.6.

2.2 Embedded systems

Embedded systems are microprocessor-based systems that are designed and built specifically to fulfill one or more roles unlike PCs which can be easily used for several purposes depending on the software used on them. This is not to say that an embedded device is always limited in its configuration as many can be configured electronically with various options as needed and have the capability to be upgraded with new software or operating systems. Embedded systems are represented in a wide variety of applications such as in home appliances like washers and microwaves, the electronic control system of an automobile or children's toys. A wide variety of processors are used, ranging from simple 4-bit microcontrollers to far more complex "system on a chip" (SoC) designs that integrate all the functions of an entire computer on a single chip as well as high powered x86 based CPUs (Heath 2003, 1-3,11.)

Embedded NAS systems are most suited to home or small office environments as while they often have lower performance capabilities than standard computer based systems they typically cost less, consume less power, usually, but not always, require less technical expertise to utilize and have a smaller "footprint" in that they take up less space. This is achieved partially by limiting the amount of features they have to what is necessary for them to perform their roles. They have no video outputs or input interfaces and have a very limited OS, usually based on modified versions of Linux or some other embedded operating system. As mentioned earlier, administration and setup of is done through the network via a web-browser although some can be modified internally to be accessible through a serial port connection. (Choubey & Singhal 2012, 134.)

Many embedded NAS systems, particularly low cost and performance models, use ARM, MIPS or PowerPC-based processor designs licensed to and produced by various companies such as Marvell. These processors are commonly used for embedded systems as they typically cost less, typically use less power than x86 based processors and are smaller due to being

more limited in function and power. The trade-off is typically lower performance although the development of ARM and MIPS processors, at least, has advanced greatly in recent years. Many now come close to or exceed the performance of similar x86 processors in certain situations although with an increase in power consumption. Improvements in the power efficiency of x86 processors proceed as well, allowing them to compete more readily in markets traditionally dominated by non-x86 processors. Embedded NAS systems that are based on x86 architectures and usually are higher in cost and performance but as stated earlier, not always. (Dang & Angelini 2012.)

Embedded NAS systems are manufactured by many companies, with more than 20 different manufacturers listed on Amazon.com alone, and are readily available. Some come with non-upgradeable storage, while others are intended for a consumer to install their own storage into. A fairly recent development in this area is that many routers now have the capability to act as a NAS device through the use of an external USB storage device being plugged into them although the performance is comparatively poor for the most part. Another recent development is the integration of an internal hard drive into a wireless router to provide a multi-functional NAS device. Also available are devices that have multiple USB ports with a single Fast/Gigabit Ethernet connection that provide network access to all USB storage devices connected to them.

2.3 Communication and Network Protocols

In order to provide file sharing, NAS systems have to be able to communicate with other computers on a network. Having support for the necessary communication and network protocols are key in doing this. Protocols are usually classified by layer such as in the Open Systems Interconnection (OSI) model developed by the International Organization for Standardization (ISO). The number of protocols that exist is very high and is continually growing. The ones most relevant to NAS are file sharing and file serving protocols which operate at layer 7, the application layer of the OSI model and TCP/IP which works at layers 4 through 2 of the OSI model (Javvin Technologies 2005, 1-2.)

File sharing and file serving protocols are defined by Smith (2004, 7) as being distinct from each other in that file serving protocols are set up where a client has to download the file, edit it locally and upload it back. They provide no means of keeping others from making changes to the files during this time. File sharing protocols differ by allowing files to be accessed as if

they are stored locally and locking access to them during this time. Some file sharing protocols also allow access to other resources such as printers. Another difference is that file serving protocols typically require a client to access the files while file sharing protocols are typically integrated into the operating system itself although file sharing clients do exist such as “smbclient” for Samba. While the term “file sharing protocol” is seen far more often in computer literature than file “file serving protocol”, both terms are well suited for describing what those protocols accomplish.

TCP/IP (Transmission Control Protocol/Internet Protocol) is a “protocol suite” or group of several protocols under a common name. It is the primary protocol suite used for the Internet as well as many other networks and it was originally developed for the U.S. Department of Defense “Advanced Research Projects Agency” (DARPA) network in the 70’s. DARPA funded further development of it at universities and their networks would form the basis of what would become the Internet. While TCP/IP consists of many different protocols, the two from which it derives its names are quite possibly the most important to it. The Transmission Control Protocol (TCP) is important as it provides a “guaranteed, connection-oriented transport system” and Internet Protocol (IP) is important because it provides features necessary for other TCP/IP protocols such as IP addressing. There is much more to TCP/IP but that is beyond the scope of this paper (Miller 2009, 3-5, 59.)

It can be said that all NAS devices and operating systems available at this time support SMB/CIFS, typically used with Windows computers, and NFS, typically used with Linux and Unix computers. Many others also support AFP which is Apple’s file sharing protocol and FTP which is a file serving protocol used over the Internet. Each of these protocols started off working other network protocols but all now support TCP/IP in some way. TCP/IP is important to NAS systems because it is the de facto network communication protocol suite and is needed for them to communicate with other networked systems.

2.3.1 Server Message Block/Common Internet File System (SMB/CIFS)

SMB/CIFS is the file sharing protocol used by Microsoft for its Windows operating systems and provides file and printer sharing. It has support for usernames and passwords to authenticate users accessing shared network resources. This authentication can be done on individual servers or through a “domain controller” which is a single computer that handles all authentications. File access can be controlled through access control lists (ACLs) and file

ownership information can be tracked. File metadata, such as file name lengths, read-only flags and other file properties specific to Windows systems, are also supported (Smith 2004, 12-13.)

Based on work published by IBM in 1984, Microsoft and Intel developed and published the proprietary OpenNET File Sharing Protocol. After Intel withdrew from development the protocol was renamed by Microsoft to SMB File Sharing Protocol, or just “SMB” for short. In 1992, SMB was made a standard protocol and no longer proprietary by the X/Open committee. According to The Open Group 1997, this committee became The Open Group in 1996 after merging with the Open Source Foundation. After further proprietary independent developments by Microsoft, SMB was renamed by that company to CIFS in a specification published in 1996 and replaced SMB fully in Windows 2000 under that name. Because it is proprietary, Microsoft has full control over how CIFS is developed. Fortunately, Microsoft allows other companies to use CIFS in their devices royalty-free so this is not a major issue in that regard but they have strict prohibitions on open-source implementations (Long 2006, 20-21.)

According to Smith (2004, 12) CIFS typically uses the NetBIOS (Network Basic Input/Output System) API (application programming interface) for network file access and TCP/IP as a network protocol. In more recent versions it can be used directly on TCP/IP without NetBIOS (Long 2006, 21). Because CIFS supports printer sharing many NAS devices can also be set up as a print server as well.

2.3.2 Samba

Samba is a suite of open source programs for Linux, Unix, and several other operating systems that provides file and printer sharing services to other computers using the SMB/CIFS protocol (Terpstra 2006). The latest stable version available at Samba.org as of this writing is 4.0.3.

In December of 1991, Andrew Tridgell began his development of an open source implementation of SMB that would come to be known as Samba. Basically, it started off as a personal project of his where he made his own packet analyzer to view packets being sent by a network running on a DOS system. From that, he deciphered what the packets did and recreated the functions in a program of his own so that he could mount shared disk space

between the DOS based network and a Unix workstation. After improving his program and releasing it to the public, it remained as is for a while until it was ported to Linux. Sometime after that he found that he had implemented the SMB protocol on his own and development continued from there to become Samba (Tridgell 1998.)

2.3.3 Network File System (NFS)

Development of the Network File System protocol (NFS) by Sun Microsystems began in 1984 and is the most widely used file sharing protocol for UNIX and Linux computers. In 1986, PC-NFS was released by Sun which provided NFS support for PC operating systems. The NFSv2 (version 2) specification was published in 1989 and NFSv3 in 1995 and both were widely regarded as standards which became official in 2000 when NFSv4 was released (Long 2006, 22.) The latest version per RFC 3530 (Request For Comments) was published by the Internet Engineering Task Force (IETF) in 2010 and known by some sources (e.g. linux-nfs.org) as NFSv4.1. Unlike CIFS, NFS is a file sharing protocol only and printer sharing must be provided by another protocol, usually through the Common Unix Printing System (CUPS) (Vugt 2008, 287).

There are many differences between the different versions of NFS although all are able to communicate with each other without much difficulty, sometimes requiring nothing more than changing an option stating what version is being mounted (Collings & Wall 2005, 299). It can be assumed that older versions of NFS are no longer used or falling into disuse due to newer versions having better security, support for larger files, better reliability, file ownership improvements, addition of new features such as support for ACLs and other improvements as can be seen at nfs.sourceforge.net.

2.3.4 Apple Filing Protocol (AFP)

The Apple Filing Protocol (AFP) is Apple's proprietary network sharing protocol used almost exclusively on their Macintosh computers. Originally developed to run on their proprietary Appletalk network protocol over a serial line, it now runs on TCP/IP over an Ethernet connection (Bartosh & Faas 2005, 376). As with other file sharing protocols, many other changes have been made to AFP over the years that have enhanced its security and reliability, improved its file sharing capabilities, file ownerships and access rights, and support for modern issues such as files over 2 GB in size and are detailed in Apple's AFP documentation

(Apple 2012). Because of it being a proprietary protocol primarily in use with Macintosh computers, many NAS devices, particularly low-cost ones, do not fully support this protocol although this may increase with increased usage of Macintosh computers.

2.3.5 File Transfer Protocol (FTP)

While FTP is a file serving protocol and intended more for transferring files over the Internet, it can be and is used over local area networks. Most NAS systems have support for it, much like traditional file servers and even though they are typically used for local area networks (LANs), NAS systems can be accessed over the Internet in this fashion. FTP came into existence in April of 1971 with the publishing of the first the first FTP standard. It was originally designed to work over the Network Control Protocol (NCP) which is the predecessor of TCP as the Internet did not exist at that time. Several more revisions of the standard were published in the following years and in 1980 a standard was published specifying how FTP would work over TCP/IP. Several more revisions have been published since, usually regarding new additions such as better security measures. (Kozierok 2005, 1170-1171)

FTP has some security issues, one of which is that it does not use encryption which means that anyone using a packet analyzer can see passwords, usernames and files sent over it. FTP packets can also be intercepted and altered by someone with fraudulent data and then sent on to their final destination in what is called a “man-in-the-middle” attack. More secure alternatives to FTP exist such as Secure FTP (SFTP) (Ciampa 2009, 421.)

2.3.6 Web-based Distributed Authoring and Versioning (WebDAV)

WebDAV is a set of Hypertext Transfer Protocol (HTTP) extensions first proposed in 1998, becoming main stream by 2002, which provides users a secure way to create, update and manage files on a web server. Additionally, multiple users can use it to edit documents jointly without worry of overwriting another’s changes as it locks files to specific users. It thoroughly integrated into HTTP, making use of many of its functions and cannot function without it. WebDAV is available in many NAS devices and operating systems for remote file editing and essentially functions as a file sharing protocol does except with the intent of being used over the Internet and not a local area network (Dusseault 2004, 2, 5, 14 -16.)

2.4 Direct Attached Storage and Storage Area Networks

Sometimes the terms DAS (Direct Attached Storage) and SAN (Storage Area Network) come up in topics regarding network storage. They are important in regards to NAS in that while these are separate concepts, DAS can be part of a NAS system and SANs can include NAS systems. SANs are particularly complex and utilize a wide variety of protocols and methods for intercommunication.

2.4.1 Direct Attached Storage (DAS)

Direct Attached Storage is almost self-explanatory in that it is storage that is directly attached to a computer without using a network. These can range from simple external USB storage devices and complex external devices that have multiple drives in them that connect through high speed buses such as eSATA or SCSI (Small Computer System Interface) to the actual internal storage of the computer itself. In regards to network storage, these would normally be attached to a server and then set up for access over the network in some way. This is a very limited way of adding storage to a network as it can be constrained by the performance of the server as more users utilize its resources. Adding additional servers is costly as well and can result in underutilized hardware as servers are usually meant for uses other than serving files (Holtsnider & Jaffe 2007, 437.)

NAS systems are a solution to this in that they provide an easy means of expanding storage on a network with a lower cost than purchasing a full server. Some NAS systems make use of direct-attached storage for expansion by having additional space for internal hard drives to be added or through external devices such as USB storage devices. That being said, most low-cost commercial NAS systems are very limited in their expansion capabilities, typically being limited to whatever storage that they come installed with.

2.4.2 Storage Area Network (SAN)

The following material on storage area networks is primarily from IBM's Redbook, "Introduction to Storage Area Networks and System Networking" (Tate et al. 2012). A storage area network (SAN) is a complex and specialized high-speed network which has a main purpose of transferring data between computers and storage devices. SANs originally used high-cost fiber-optic cabling for network connections but developments in technology

now allows usage of lower cost copper wiring based solutions and equipment such as Gigabit Ethernet. The high speed of SANs is very appealing to larger companies that can more readily utilize the benefits while being able to afford the higher costs.

A SAN typically has direct connections to all devices on it which allows for server to server, server to storage and storage to storage communication. Direct connections to storage devices allow multiple servers to utilize them without going through another server's I/O bus. In this way, storage devices can be centralized and consolidated instead of spread out among multiple servers. SAN storage devices and servers can also be located far away over great distances, allowing for remote data storage that can be used, for example, as backup in case of a natural disaster. High-speed server to server communication allows for servers to be clustered which can be beneficial in such things as allowing computing loads to be spread across multiple servers.

SANs usually use block data transfers via the Fibre Channel (FCP), Fibre Channel over Ethernet (FCoE), Fibre Channel over IP (FCIP), In or Internet Small Computer System Interface (iSCSI) protocols, all of which are based in some way on the based on the SCSI I/O protocol. While NAS devices are set up for file transfers through SMB/CIFS or NFS, it is possible to attach a NAS system to a SAN which treats it the same as any other server.

Regardless of the method or topology used for transferring data, the terminology for the various parts of SANs is the same for all. Information is sent between "nodes", one of which is called the "transmitter" or "initiator" which is the source of the information and the receiving node is called a "receiver" or "target". Nodes can be any device that connects to a SAN that sends and receives information, such as a server or storage device.

2.4.3 SAN Protocols and Architectures

Fibre Channel is the most often used technology for SANs and primarily uses fiber optic for interconnecting devices but can operate over copper cabling as well although with lower speeds and more limitations. Originally intended as a backbone technology for local area networks (LANs), it was instead developed as an alternative to Serial Storage Architecture (SSA), an alternative technology to SCSI developed by IBM, and came to market in 1997. Components are separated into Base2 and Base10 varieties with the original speed for Fibre Channel being 100 megabytes per second (MBps) in one direction and was named 1GFC.

Base2 components are numbered in bases of two, starting with 1GFC and doubling each speed increase to a current maximum of 1600 MBps named as 16GFC with even higher speeds planned for the future. Base10 components start with 10GFC which gives speeds of 1200 MBps, and doubles each time to the current maximum of 40GFC or 4800 MBps. The largest downside to Fibre Channel is that it requires specialized hardware which is typically more expensive than more common Ethernet hardware (Troppens et al. 2009, 66-67, 71-72.)

Fibre Channel is also the name given to the protocol used most often on Fibre Channel SANs and is sometimes abbreviated as “FCP”. As mentioned earlier, FCP is based upon the SCSI protocol but with the intent of utilizing it over a network instead of SCSI cabling. The SCSI protocol is mapped onto the Fibre Channel network where there are some differences. SCSI uses parallel data transmissions with multiple devices attached in a daisy chain while a Fibre Channel network transmits data serially which allows for higher speed and longer cable lengths. The logic used for scanning devices and arbitration work completely different on the two as well. There are likely other minor differences that must be taken into consideration as well. These differences are all handled by the FCP driver. Other protocols that run on Fibre Channel networks are IPFC which is used for transferring IP packets and FICON (Fibre Connection) which is used for mapping the ESCON protocol used by mainframes on to a Fibre Channel network (Troppens et al. 2009, 67, 87-88.)

The **Internet Small Computer System Interface (iSCSI)** protocol works very similar to Fibre Channel in that it too uses the SCSI protocol but it maps it through the TCP/IP protocol running over an Ethernet connection instead. iSCSI was standardized in 2003 and is becoming more prevalent in working environments because it can utilize lower cost Ethernet based networks while requiring nothing more than drivers to implement the iSCSI portion on a computer’s cards. A specialized iSCSI card called a host bus adaptor (HBA) which implements the iSCSI protocol in its hardware can also be utilized instead to reduce the load on the computer’s CPU. For an iSCSI SAN to communicate with a Fibre-Channel one, an iSCSI-to-Fibre-Channel gateway is required as the two protocols are different from each other even though they both make use of the SCSI protocol (Troppens et al. 2009, 105-106.)

Internet Fibre Channel Protocol (iFCP) differs from iSCSI in that instead of mapping the SCSI protocol over TCP/IP, iFCP maps the entire Fibre Channel Protocol onto TCP/IP. The benefit of this is primarily to those companies who have already invested heavily in Fibre Channel devices yet wish to utilize a lower cost IP/Ethernet network infrastructure with them.

To do this, switches must be able to provide a connection to a Fibre Channel network or alternatively a FCP-to-iFCP gateway can be used. Also in existence is mFCP (Metro FCP) which is essentially the same as iFCP except that it works with UDP packets instead of TCP packets for higher speed with less reliability. iFCP/mFCP were accepted as standards in 2005 but do not seem to be coming into major use due to the complexity of the protocols and lack of major benefits over other protocols (Troppens et al. 2009, 106-108.)

Accepted as a standard in 2004, the **Fibre Channel over IP (FCIP)** protocol works differently by encapsulating Fibre Channel frames in TCP/IP packets and is known as a tunneling protocol. It is used most often for creating a point-to-point connection between two Fibre Channel SANs over a wide area network (WAN). In this way, long distances between SANs can be overcome utilizing a standard network such as that used for Internet and, for example, can be used for making backups in case of a large-scale catastrophe. Another benefit is the possibility of encrypting the data sent through IPsec (Internet Protocol Security) (Troppens et al. 2009, 108-109.)

Fibre Channel over Ethernet (FCoE) works in an opposite way from iSCSI, FCIP and iFCP in that it uses Ethernet as its network technology for the Fibre Channel protocol while bypassing any use of TCP/IP. It does this by encapsulating each Fibre Channel frame one for one in an Ethernet frame which keeps them from being fragmented. This encapsulation and subsequent decapsulation on the receiving end is done by a very simple protocol which allows for low-complexity implementation in hardware. Since it transfers Fibre Channel packets in such a way, Fibre Channel equipment can be connected to such a network with far less issues. One problem with FCoE, however, is that it requires specialized equipment that supports the larger Ethernet frames needed to encapsulate the Fibre Channel frames and can interpret the Fibre Channel protocol (Troppens et al. 2009, 124-127.)

In regards to NAS, the majority of if not all of the NAS operating systems that are used in standard computer based NAS systems support iSCSI in some form or another as it uses standard Ethernet equipment and TCP/IP. With embedded NAS systems, the higher end devices typically support iSCSI while lower end ones do not. In both types, it is merely a matter of providing proper driver/operating system support for the protocol. Other SAN protocols require specialized equipment to be installed though and this is usually only possible on the standard computer based NAS systems. Embedded NAS systems would have

to be prebuilt with other network types in mind and which would incur additional costs in purchasing them.

2.5 Wake-On-LAN (WOL)

One possibility to limiting the on-time of NAS devices is to set them up to utilize wake-on-LAN (WOL) which is a way of “waking up” a system that is in standby mode through a network connection. In a small office or home environment such a feature could present reasonable savings over a year, particularly for custom-made NAS systems that utilize standard off-the-shelf computer parts which typically consume more power than low-cost embedded systems.

Wake-on-LAN allows a network interface card (NIC) to bring a system out of standby mode when it receives a specialized packet. In addition to being the original type of packet first used for WOL, the most commonly used is called a “Magic Packet” which is a type of broadcast frame. The data payload of the frame contains six bytes of all ones which is FF repeated six times in hex format, followed by the network interface card’s MAC address repeated 16 times (Held 2012, 208-209.)

Packets can be a unicast transmission that is targeted at the computer to be woken up or a subnet-directed broadcast, such as the “Magic Packet” mentioned earlier, that will be sent to all computers on that subnet. The network card and computer (usually through a BIOS setting) must have support built into them for this feature to work on a LAN. If routers and switches are involved, they too must support WOL in some way for it to work properly between different subnets (Amaris et al. 2012, 81.) Windows, Linux, OS X and FreeBSD all support WOL. The ease of setting each up for it varies and going into detail about each is beyond the scope of this paper.

3 NAS OPERATING SYSTEMS

As explained earlier, a wide variety of NAS operating systems exist. Technically any operating system that can be remotely administered and set up with file shares can be used for this purpose but “NAS operating systems” are specifically set up for this in that they typically have only those services and protocols necessary for providing network file sharing running.

That being said, some can take up additional roles, such as being a virtual machine host or as a web server, as necessary.

Embedded NAS systems typically use modified versions of Linux or some other embedded operating system. Some of these operating systems are also known as a RTOS (Real-Time Operating System). RTOSs differ from normal operating systems in that they are designed to respond to inputs and events within a defined time although some normal operating systems work this way as well. This is very important in network devices and other systems where a delay could cause timeouts and errors (Heath 2003, 220.) Some companies also develop their own proprietary operating systems, some of which are based on Linux or FreeBSD (e.g. LaCie).

The following subchapters will cover in more detail the various operating systems commonly used by NAS systems. All of these operating systems have support for the protocols from chapter 2 in some form or another, sometimes full support, other times limited.

3.1 Linux, FreeBSD, illumos and Other “Unix-like” Operating Systems

Linux is a “Unix-like” operating system that offers easy customization for many purposes as it is open source which means that its source code is available to anyone who wants to use and modify it. Linux is the “kernel” or core of the operating systems based on it in that it only provides most simple functions such as memory and process management. From this kernel, many “distributions” have been developed which build upon the kernel with the various programs needed to fulfill the functions of a full operating system (Raggi et al. 2010, 5.)

The same hold trues for FreeBSD and illumos (a fork of OpenSolaris). Like Linux, they are both “Unix-Like” operating systems and are freely available open source kernels and according to Matzan (2007), FreeBSD even supports binaries compiled for Linux if it is properly configured. Unfortunately, OpenSolaris is no longer being updated under that name since Oracle acquired Sun Microsystems who was the main developer of that operating system and made it closed-source. Fortunately, an open-source fork of it exists under the name of “illumos” which is still undergoing active development although illumos lacks the hardware support that existed in OpenSolaris and may not work on properly on systems that the older operating system did (Germain 2012).

For the purposes of custom built NAS systems, one can either find a modified version of one of the previously mentioned operating systems already optimized for NAS usage or take a standard distribution and repurpose it as needed. Depending on the file system installed and other options, they all can be suitable operating systems for older systems although this is dependent on hardware support of which Linux seems to have the highest, followed by FreeBSD and lastly illumos. That being said, support for newer hardware can be lacking although this has improved compared to ten years ago and installation of drivers can be complicated for some, with difficulty varying even among different distributions of the same operating system kernel.

3.1.1 OpenFiler

One NAS distribution of Linux is OpenFiler. As of this writing, the latest version is 2.99 which was released in April of 2011. It is based on rPath Linux which is based on Fedora Linux. It is somewhat commercialized as while it is freely downloadable, community support is somewhat lacking, the official manual requires payment to be downloaded, and company support requires payment. Certain enterprise targeted features also require payment (Openfiler 2012.)

rPath, the company responsible for developing rPath Linux, was bought out by a company called “SAS” in November of 2012 so future development of its Linux branch is unknown (SAS 2012). Currently the rPath website is no longer working and at least one site marks it as “no longer in development” (DistroWatch 2012). Because of this, OpenFiler may have development issues in the future. That being said, it might not be hard for those knowledgeable in Linux to update it on their own.

3.1.2 OpenMediaVault

Based on Debian Linux, OpenMediaVault is still in development as of the time of this writing. The latest version is 0.4 which was released December of 2012. There is a well-developed wiki on it and community support is strong. Official documentation is somewhat complicated but freely available. It is completely non-commercial but accepts monetary donations for support and development (Theile 2013.)

3.1.3 Linux + iSCSI and Other Additions

As mentioned previously, it is possible to take a Linux or other OSs' distribution and modify it for NAS purposes. By doing so, one has more control over what services are running and what gets installed. One can also reduce the footprint of the OS this way by deleting unnecessary files and services as well.

For example, Ubuntu is a very popular and comparatively easy to use Linux distribution based on Debian, another Linux distribution. To enable iSCSI, NFS and SMB/CIFS as needed by the customer, "packages" which contain all the programs needed to support the protocols require installation before they work properly (Ubuntu Server Guide 2012, 225, 227, 288). To remotely administer the system, one can use SSH (Secure Shell) to access the system or enable "Remote Desktop Viewer" if it has a GUI available (Wallen 2010). An additional benefit would be the capability to add in extra features that may be beneficial such as "GlusterFS" which is a "meta-file-system" in that it builds its file system on top of the file systems of the devices it stores data on, basically combining several different storage devices into a single source (Layton, 2010). By a combination of these, one can build a NAS system that would suit the purposes of many users, however, it can be very complicated and beyond the skills of most casual users.

3.1.4 NexentaStor

NexentaStor is based on the illumos OS forked off from OpenSolaris. It has a "community edition" for free usage and a commercial variant with support for additional features such as storage of more than 18 TB (terabyte or 1000 gigabytes). The latest version as of this writing is 3.1.3.5, released 22 October 2012 (Nexentastor 2010.)

3.1.5 FreeNAS and NAS4Free

FreeNAS originally started out as a volunteer developed modified version of FreeBSD in 2005 but was discontinued by the main volunteer developer in 2009. At this point, iXsystems stepped in and continued development of it. The entire project was more or less restarted from scratch and based on an embedded distribution of FreeBSD called NanoBSD. This allowed them to go to a modular design that allows for plug-ins to be developed for it so that if a

feature isn't available, one could be added easily. The latest release as of this writing is 8.3.1, released 20 March 2013 (FreeNAS 2013.)

When FreeNAS development was taken up by iXsystems, other people continued development on the original version of it. Due to naming restrictions, this version had to be renamed to NAS4Free, the latest version of which is 9.1.0.1, released 5 February 2013 (NAS4Free 2013.)

3.2 Windows Based Operating Systems

Microsoft has a NAS version of its Windows Server operating system called "Windows Storage Server" of which the latest release is "Windows Storage Server 2012". It is not available for public purchase but it is available for evaluation by downloading from Microsoft. Basically, it is a specialized version of Windows Server 2012 with fewer features intended for the commercial NAS market (Microsoft 2013.)

Other versions of Windows Server also work very well for NAS purposes as do older Windows operating systems that can provide the needed network shares such as Windows XP and Windows 2000 and these might be a viable option for older hardware for some users who feel daunted by the task of dealing with a non-Windows operating system.

3.3 OS X Server

One alternative almost exclusive to Mac users is Apple's Unix-based OS X Server. While not specifically meant for NAS usage, it has support for many of the features common to NAS systems and is billed as being "... so easy to set up, who needs an IT department?" which may appeal to those with less technical expertise as well as those looking for something that is optimized for usage with Mac computers. The latest version is sold as an add-on to OS X Mountain Lion (Apple 2013.)

3.4 Operating System Comparison

Comparing different operating systems is by no means simple. There are many differences between each as different companies and people are involved in the programming and production of them, all with specific aims in mind that vary from operating system to the

other. All have the same goal in mind however in that they provide an easier means for a user to interact with the hardware of a computer.

One important factor for many when choosing an operating system to use for a NAS system is the price. Linux, FreeBSD and Illumos variants are all free although some require payment for certain features or technical support. Upgrading to OS X Mountain Lion is \$19.99 as of this writing and upgrading to the server version from there is an additional \$19.99 which should be appealing to those using Mac hardware. However, one typically pays a premium on Mac hardware and the upgrade is only available if you have a previous version of OS X installed. Windows Server 2012 is the most expensive with multiple versions at differing price points and capabilities available, the lowest priced starting at roughly \$501 and the most expensive at \$4,809 for single copies although these prices can vary depending on vendors and through volume licensing.

As different operating systems have different hardware requirements, it is most convenient to summarize them in table 1.

Table 1: OS Hardware Requirements

Operating System	Minimum CPU	Minimum Memory	Minimum Space for OS
Openfiler 2.99	x86 or x64	256 MB	1 GB
OpenMediaVault 0.40	i486 or x64	1 GB	2 GB
Ubuntu Server 12.10 [1]	x86 or x64 300Mhz/1 Ghz	256/512 MB	700 MB/1 GB
Nexentastor 3.1.3.5	x64	1 GB recommended	10 GB
NAS4free 9.1.0.1	x86 or x64	512 MB	128-400 MB
FreeNAS 8.3.1	i486 or x64	24 MB (8GB+ if using ZFS)	150 MB
Windows Storage Server 2012	x64 1.4 Ghz	2 GB	60 GB (Installed footprint far less than this)

OS X Server “Mountain Lion”	Not listed - varies by system manufacture date	2 GB	10+ GB plus Base OS (8GB)
--------------------------------	--	------	------------------------------

These are the latest requirements as according to each operating system’s respective websites and as can be seen, Windows Storage Server has the highest requirements while the lowest requirements are on one of the FreeBSD operating systems. x86 means any 32-bit Intel compatible processor, x64 is for the 64-bit versions and i486 is any Intel compatible processor of 486 processing power equivalency or higher.

The file systems supported by each varies greatly but goes along with what the parent operating system supports. In regards to NAS systems, differences in file systems matters less than one would think as they all appear the same to computers connecting to them over a network. More important is speed and data integrity and as can be expected, these vary greatly from one file system to another. While some operating systems, particularly Linux, have greater support for a wide variety of file systems compared to others, operating systems typically have certain file systems that are recommended by the manufacturers and are listed in table 2. While these file systems are primarily used with these operating systems, it does not mean that others cannot make use of them. Programs exist that allow most operating systems at least read-only access to non-native file systems.

Table 2: Common Operating System File Systems

Operating System	Recommended file systems per manufacturer
Linux (Ubuntu)	ext3/ext4, XFS
FreeBSD, illumos	UFS/UFS2, ZFS
Microsoft Windows	NTFS, ReFS
Apple’s OS X	HFS+

All of the file systems listed above have support for various forms of error checking and correction as well as varying support for numerous other features such as data deduplication and file compression. Data deduplication is typically used when data is repeated often such as hosting multiple virtual machines using the same operating system on the same drive or for email servers where many accounts could contain the same attachment while file compression is more useful for files that do not get repeated but can be compressed.

One relatively new file system that has been receiving a large amount of attention is ZFS which has been designed with maximum data integrity and speed in mind. This comes with steep system requirements and is recommended more for higher end NAS systems in environments where data integrity is most important. For example, the FreeBSD Handbook recommends a minimum of 1 GB of RAM plus 1 for each TB of storage space while FreeNAS recommends a minimum of 8 GB of RAM plus 1 for each TB of storage space when using ZFS RAID and an additional 5 GB of RAM for each TB if deduplication is used. If Active Directory is used with it, FreeNAS recommends an additional 2 GB of RAM for that alone.

As can be expected, this adds up so that, for example, in a computer with three hard drives of one terabyte each using RAID-Z1 (roughly equivalent to RAID 5), Active Directory and deduplication would require roughly 22 GBs of RAM. The benefit of course would be higher data integrity and speed compared to standard RAID-5. Because of these steep requirements, it is not recommended to use ZFS for most lower-end NAS systems as they typically do not have the memory needed for optimal performance, especially if using RAID.

4 INSTALLING, TESTING AND FINAL IMPLEMENTATION

Preliminary installing and testing was performed on a PC with an AMD64 dual-core processor and 4 gigabytes of RAM. The x64 (64-bit) version of each OS was used when available. Gigabit Ethernet was used for the test network as Fast Ethernet was easily overwhelmed by any amount of file transfers. Some small tests were also made with an older Pentium 4 with Hyperthreading and 2 gigabytes of memory to see how well older hardware utilizing x86/32-bit versions compare to newer hardware but these proved to be too time consuming to pursue further. Hard drives used were single 120 gigabyte and 80 gigabyte SATA hard drives and two 80 gigabyte UDMA 5 PATA drives.

No tests were made of Apple OSs as they are specific to Apple hardware only and as such were not available for testing on. The operating systems tested were OpenFiler, OpenMediaVault, Nexentastor, FreeNAS, NAS4Free, Ubuntu 12.10, Linux Mint 14, Windows 7, 8 and Server 2012.

4.1 Installation Details

All operating systems except for NexentaStor, OpenMediaVault and Windows Server 2012 installed without any issues. Nexentastor installation locked up when done from a USB storage device such as a USB DVD drive, and OpenMediaVault installation will sometimes lock up if the serial port on a PC is enabled. Furthermore, Nexentastor was never able to reboot upon installation despite leaving the system on for over an hour. It is suspected that this was due to a hardware incompatibility and due to time constraints this was not further investigated beyond reinstalling the operating system multiple times with different installation media. Windows Server 2012 had a small error where if a driver disk is in the computer instead of the installation disk, you will sometimes receive a message of “Windows can’t be installed on this drive.” with an error code of 0x80300001 when trying to select a drive for installation. This was solved by placing the installation disk back into the computer but the error gave no indication that this would resolve it.

Most minor issues mostly centered on setting up partitions and groups for file sharing. The Linux based operating systems have an auto-partition option which if used, installs the operating system to multiple drives. This may pose an issue if those drives are meant to be used for specific purposes later on and it is the recommendation of this author to either set up the partitions manually or have only the drive or drives to be used by the operating system connected then connect storage drives later. Windows also had this problem to a certain extent in that it insists on creating a small 100 to 350 MB partition, dependent on the operating system installed, that will sometimes be installed to the primary operating system drive and other times to secondary drives. In both cases, these extra partitions interfered with setting up software RAID by preventing the use of it entirely or causing a mismatch in drive space availability.

One issue with the FreeBSD based operating systems is that they do not allow the system drive to be used for shared storage later and it is recommended by the authors of them to install them to a flash drive if possible. Another issue that occurred with Unix-like operating systems was that pre-existing partitions were not always recognized by the remote administration browser and required changes from the command prompt on the physical server.

Other than the previously mentioned issues, setting up partitions was straightforward for all systems. Setting up software RAID was easily done as well except on the Ubuntu and Mint Linux installations where it was far easier to do when installing the operating system than

later. This is due to the installation program easily allowing for creation and setup of partitions for RAID while doing this afterwards requires use of a command line program called “mdadm” that can be complicated.

The steps for setting up SMB/CIFS shares were nearly identical for each of the Unix-like NAS operating systems tested. All required starting of the SMB/CIFS services then entering of a network (NetBIOS) name for the NAS system. User accounts had to be created although some came with set up with guest accounts already made. These accounts then had to be assigned to access groups that had proper permissions for accessing the shared directories and the shared directories had to be set up with permissions for those groups that have access to them.

Connecting to an SMB/CIFS share through a Linux based operating system such as Ubuntu requires a file manager with Samba support such as “Nautilus” or “Nemo” and accessing NFS shares requires a package called “NFS-common”. As these may not be pre-installed on all distributions, it may be necessary to have Internet access so that the packages can be downloaded and installed as needed.

4.2 Testing and Comparisons

After installing the operating systems and setting up various groups and user IDs for SMB/CIFS and NFS, the systems were tested to verify that the file-sharing was working properly. SMB/CIFS was straightforward as Windows and Linux both have support for it. Windows found the SMB/CIFS shares for all systems with no issue while Linux required installation of the Nautilus package as the default Nemo file manager had difficulties in finding the shares. Connection testing was done using clients using Ubuntu, Mint Linux, and Windows 7 and 8.

However, testing NFS shares proved to be more difficult as Microsoft has dropped built-in support for the majority of their Windows operating systems starting with the release of Vista, having it available only in their server editions, the Ultimate and Enterprise versions of Vista and 7 and Windows 8 Enterprise edition. The server versions have support for creating NFS shares and connecting to them while the Ultimate/Enterprise versions have support only for connecting to them. While third-party software exists that brings NFS functionality to Windows systems that do not have built-in support for it, it is recommended by numerous

sources to instead use SMB/CIFS for file sharing between Unix and Windows systems. Despite this, setting up NFS on those Windows systems that support it natively was simple and without errors and worked reliably during further testing.

When comparing NFS and SMB/CIFS, there were some differences in performance depending on the operating system being connected to the shares. Both easily hit the bandwidth limits on a Fast Ethernet connection with speeds for both averaging 10.5 to 11.5 megabytes/second speeds on all systems tested. On a Gigabit Ethernet connection, the speeds varied a bit more.

For example, on one system where Linux was tested with a connection to a NAS server, NFS proved to be faster for transferring files to the system, averaging 37 megabytes/second versus 30 for SMB/CIFS. However, that same computer with Windows Server 2012 averaged 14 megabytes/second for NFS and a far faster 50 to 55 megabytes/second for SMB/CIFS while connected to the same NAS server. Interestingly, SMB/CIFS performance for all operating systems consistently averaged the same 50 to 55 megabytes/second on the same hardware when connected to the same Windows PC client although there were a few times where memory caching was evident and speeds over 100 megabytes/second were obtained.

Memory caching with SMB/CIFS was most evident on the final implementation hardware where a direct Gigabit Ethernet connection repeatedly hit the maximum bandwidth transfer rate of 118 megabytes/second after transferring the test files at least once while a NFS share and iSCSI connection between the two both averaged only 50 megabytes/second regardless of the times previously transferred. NFS performance was contradicted, however, by further tests on the same system where a total of twelve different clients downloaded the same test files at a speed of 10 megabytes/second each or 120 megabytes/second which is far higher than the single direct connection.

As all of the operating systems tested support software RAID, an attempt was made to test that as well. For this, the two matching older 80 GB hard drives were used. The read speeds for the drives when set up in mirror or striped RAID were near identical on all systems tested and did provide some amount of a performance increase. Mirror RAID or RAID 1 basically copies the data to drives at the same to provide better file integrity and increased read speeds while striped RAID or RAID 0 interleaves the data amongst both drives in an attempt to increase read and write speeds. Testing of RAID was done over the network via SMB/CIFS to

provide comparison with non-raid performance. Read speeds averaged 50 to 54 megabytes/second compared to 40 per second for the single drives with writing speeds around 45 megabytes per second for both types of RAID. On single drives, the performance varied in disparate amounts with 35 megabytes/second write speeds for the same drives on Windows Server 2012 and on the Unix-like NAS operating systems, write performance was oddly higher on some with the single drives consistently reaching 50 megabytes/second on write speeds and it is suspected that memory caching again skewed the results in most instances.

iSCSI was tested to verify that it worked as well and provided speeds similar to that of SMB/CIFS on Windows clients connected to the NAS systems. What proved to be most interesting was the setting up of two iSCSI shares in RAID on the iSCSI initiator system. This method provided the fastest consistent read speeds from the servers of an average of 60 to 68 megabytes/second. It is suspected that memory caching at some point may have skewed the results in this case as well.

In regards to the small amount of tests made with the 32-bit operating systems on the older Pentium 4 system, these were consistent as well, averaging 32 megabytes/second performance read and write over SMB/CIFS with the same drive used for testing the 64-bit systems and it achieved 50 megabytes/second with software raid. Testing of NFS was not done on this system.

A brief mention will be made of the NAS storage capabilities of the Gigabit router used during testing. It has the capability for attaching a USB storage device to a USB 2.0 connector on it and allows sharing of this storage device on a network as a SMB/CIFS share. The speed of this proved to be lackluster though, averaging roughly 4.5 to 5 megabytes/second when transferring data to and from it. The same USB drive when plugged directly to a PC averaged 30 megabytes/second with the same test file. While these speeds are not likely representative of all router based NAS system, it was not promising and promptly dismissed as a plausible alternative.

With SMB/CIFS used on two teamed Gigabit Ethernet network cards on the final server equipment with Windows Server 2012, a speed of 1.4 gigabits per second (Gbps) or roughly 175 megabytes/second was obtained regularly through a switch connected to 21 other PCs utilizing 100 megabit per second (Mbps) connections. This was sustained while transferring the same files to another PC through a dedicated Gigabit Ethernet connection averaging 350

Mbps or 45 megabytes/second. NFS hit the same limit of 1.4 Gbps on the same system with the same setup as SMB/CIFS although without the additional single PC connected via a dedicated Gigabit Ethernet connection as that specific PC did not support NFS connections. The server outgoing speeds and incoming speeds to the clients held steady during the transferal of large files. With the transferal of smaller files there was much more variation in speeds on the client side but server outgoing speed remained constant at the 1.4 Gbps rate on the linked network cards.

The reasons for the similarities in performance are likely due to limitations of the hardware used during testing. In the case of the dual-core 64-bit computer's performance versus the hyper-threaded single-core 32-bit computer's, the limiting factor is likely to have been the SATA hard disk drives used for testing as both were 5400 rpm drives meant for general use. Caching by the operating systems also played some part in their performance and it is reasonable to conclude that increased memory would have improved benefits in this regard. The 1.4 Gbps limit with two teamed network cards could have been a result of the switch used or it could have been an issue with how network card teaming worked on the Windows 2012 server. The differences in NFS versus SMB/CIFS performance seem to be largely operating system dependent with NFS performing better for Linux Clients and SMB/CIFS for Windows clients.

4.3 Final Implementation

The original intention was to set up a NAS system using a Linux based NAS operating system for file shares. It would have support for SMB/CIFS, NFS and iSCSI. The system would be set up on a HP computer with an Intel Core Duo CPU and 8 GB of RAM and utilize a solid state hybrid drive (SSHD) for storage. A single Gigabit Ethernet connection would provide a connection to the server to the rest of the network.

It was found early on that the SSHD has driver support for Windows only as it was not meant for server/enterprise usage. This SSHD is a 1 terabyte hard disk drive mounted on a PCI-E x4 card with an integrated 100 gigabyte solid state drive (SSD) and is manufactured by a company called OCZ. It is intended by the manufacturer that the SSD be used as a cache but it was found that the caching software, Dataplex by NVELO, is supported only in Windows 7 although Windows 8 support is supposedly in the works.

Because of the driver and software caching issues, the decision had to be made to use Windows Server 2008 or Server 2012. Windows Storage Server 2012 could not be used as it is intended for evaluation purposes only and is only available to NAS manufacturers. Server 2008 had the benefit of supporting the caching software while Server 2012 had the benefit of newer features, improved reliability and improved SMB/CIFS and NFS capabilities. The decision was then made to use Server 2012.

It was then decided to use the solid state portion of the drive as the system drive and set up the file shares to the hard drive portion. Solid state drives are beneficial in that they have greatly increased read and write speed over traditional magnetic hard drives and no moving parts that consume more power, are prone to physical damage and can wear out over time. This comes with an increased cost, less storage capacity and an issue with long term usage. The long term usage problem is due to the fact that the storage of a solid state drive has limited number of times it can be written to although this is a fairly high number with recently released drives.

One specific issue concerning the solid state drive used for the final implementation is that it is not recognized as one by Windows, likely due to being mounted on a PCI-Express card and showing up in the Device Manager as a SCSI device. Because of this, certain features meant to increase the functionality of solid state drives may not have been enabled, most specifically that of TRIM.

TRIM is a command sent to a solid state drive that tells its controller that deleted blocks of data are no longer being used and can be overwritten by the controller with empty data. If this is not done regularly, the write speed of a solid state drive decreases over time. This happens basically because solid state drives do not erase their data in a way that their controllers recognize it as being freely available for writing to later. If a solid state drive happens to need to write to a location that had been written to previously and not emptied by the TRIM command, it has to copy that entire block to a cache or other temporary holding location, modify the block, then rewrite the entire block to the drive (Shimpi 2009.)

Whether or not the lack of TRIM will be detrimental to the server will remain to be seen. A utility to confirm it was working showed that it was not although Windows has a command that shows it is at least enabled on the system. By entering “fsutil behavior query DisableDeleteNotify” at a command prompt with admin rights enabled, a result of “0” indicates that TRIM is enabled while a “1” means that it is not. On the final server this came

up as “0” but this does not mean anything as it also came up with the same result on numerous other systems without solid state drives.

Another issue with the solid state drive not being detected properly is the scheduling of the disk defragmenter on it. Solid state drives do not access their data in the same way as traditional hard drives so defragmenting them is not needed. Also in defragmenting a solid state drive, data is deleted and rewritten to it which wears the drive out faster and results in the previously mentioned issue regarding deleted data.

Attempts were made to use a different software caching program called VeloSSD published by a small company named Elitebytes. Early tests using this software were very promising with it using a small portion of the solid state drive to cache the hard drive. Tests showed that the caching software did perform as promised as activity for the hard drive dropped to near zero while the solid state drive showed continual activity while transferring files over the network. This was monitored via the Windows Resource Manager. Unfortunately, the software cache encountered a problem upon reinstallation of the operating system that prevented the system from booting whenever the cache was enabled. Despite reinstalling the operating system multiple times, this issue was never resolved. It is suspected that there may be some residual data on the drives causing the error but deletion of all partitions and complete reformatting of the solid state drive failed to resolve the issue.

Roles installed for the server were as follows:

- File server
- Data deduplication
- Server for NFS
- iSCSI target server
- File Server Resource Manager

Installing of these roles was necessary for the server to fulfill certain purposes that the customer required of it.

The usage of iSCSI proved to be problematic in that the original intention was to use the entire hard drive as an iSCSI target that would be accessed by multiple computers. This did not work properly due to iSCSI targets being treated as a directly connected hard drive in regards to multiple iSCSI initiators accessing the same target. This results in the data on the target becoming corrupted as the initiators have no way of tracking what changes the others

have made to the target. Because of this, it was decided instead to create two 10 gigabyte iSCSI targets on the hard drive that would be used for demonstration purposes.

A single SMB/CIFS and NFS share directory was created on the hard drive using the Server Manager. The same directory was used for both with the intention to use it for sharing operating system installation disc images over a network. A user account named “Student” was created and given read only access to the directory. A small additional step in doing so required temporary disabling of the “required password complexity” security so that the simple password for that account could be used. Both shares were tested from a client computer to verify that they could be read from but not written to.

The hard drive also had data deduplication set up on it with deduplication to take place only with files more than five days old. Scheduling of deduplication was set for Saturday and Sunday so as to not interfere with normal operation of the server during the week. At one point, an additional Gigabit Ethernet card was installed to test out how well network card teaming worked. As this proved to be beneficial, the decision was made to leave it installed in the server. In keeping with the original aims of the server being a NAS system, remote desktop administration was enabled which allows a remote user with proper authentication to administer the system from another computer on the same network.

5 CONCLUSION

In doing this thesis, it was hoped to find the best NAS operating system solution to use for standard computer hardware by comparing different working versions of each. However, all, excepting Nexentastor, proved to be so similar in performance that making a decision based on that factor was not feasible. It is very likely that with different hardware, differences would have been found with the operating systems that would have made it so though. Feature wise, all were very similar in function as well, providing NFS and SMB/CIFS shares as well as iSCSI target functionality. Software RAID was another common feature that would have been interesting to test further but since it was not required by the customer, this was not done in depth. Perhaps of particular note is the performance of NFS with a Linux client versus SMB/CIFS with a Windows client. Both performed decidedly better with those clients and it is recommended that if in an all Windows or all Linux environment to use the protocol native to it. However, in a mixed environment, SMB/CIFS performance was close enough to NFS on Linux systems that it would be the best solution for such a situation. In the end, choosing a

NAS operating system largely comes down to system requirements, hardware and technical support, cost and extended features in regards as to which one would be preferential.

Disregarding the operating system environment, the only other recommendation I can safely make in which to use would depend largely on the customer's preferences. If they are comfortable with Unix-like environments and/or prefer something low cost or free, any of the freely available alternatives will likely fit most needs. As I was able to get all but Nexentastor to work without any major issues, the main concerns would likely be that of long term stability, available technical support and documentation, and hardware compatibility issues.

While it was not possible to implement the NAS system in the manner the customer originally requested, a solution fulfilling the requirements of the customer was found and implemented. Although it did not utilize a devoted NAS operating system based on Linux as originally planned for, the usage of Windows Server 2012 for the final system was found to be satisfactory and may in fact prove to be the best choice later on due to the support provided by Microsoft for this operating system.

The largest factor in deciding to use Windows Server 2012 was the usage of the solid state hybrid drive. As it has driver support for Windows only, it severely limited the operating system choices and another limiting factor was not being able to utilize the solid state drive as a cache for the hard drive as the manufacturer intended. Utilizing the solid state drive as the system drive instead has proven to be very fast though and it may be prove to be useful in this manner instead as usage of the solid state drive for anything else is likely to be very limited. Additionally, the drive was released less than two years ago yet has already been moved to legacy hardware support on the manufacturer's website. Utilizing the solid state drive as the system drive instead has proven to be very fast though and it may be prove to be more useful in this manner instead.

Work on this thesis began at the beginning of February of 2013 although the majority of it was done in the months of March and April and it was finished in May. During the course of it, quite a bit was learned on my part about the various aspects of NAS systems such as what a NAS system actually entails as well as the essentials of how they operate. Also of interest were the details on the various protocols used by NAS systems and for storage area networks. It was also interesting working with different operating systems than Windows although there were times when setting up the Linux client operating systems for testing proved to be

stressful due to a lack of many features not being preinstalled whereas a Windows installation would include them by default.

Of the operating systems tested, I found Windows the most comfortable to deal with but this is likely due to past familiarity with that operating system. Linux proved to be the most frustrating but was workable in the end. FreeBSD seemed to be very similar to Linux but different enough in various ways as to make it interesting. illumos looked very similar to FreeBSD but I was unfortunately unable to get the NAS operating system that utilized it to work and due to time constraints did not try it on other hardware either. If I were to set up a NAS system for my own personal use, it would likely make use of Windows, partially due to my familiarity with it and also due to the fact that all of the computers in my household use some variant of that operating system. That being said, it would be interesting to try out the others for some length of time to see how they work as well as to test the long term stability of the system.

One of the largest difficulties in completing this thesis was trying to figure out how much information was enough versus too little or too much. Even when looking back at what is completed so far, it seems that some areas may be lacking but when thinking about what to add, it occurs that other areas must be expanded in kind. The lack of conclusive material in regards to NAS systems themselves did not help either. While many articles are available online in regards to them, most are focused on commercial NAS devices and neglect to mention any aspect of the “do-it-yourself” operating systems available for normal computer hardware. Some articles are overtly commercial and are nothing more than advertisements for specific commercial NAS devices while others provide contradictory information and still others are so poorly written as to be unusable for the purposes of this thesis. Printed materials detailing NAS systems were also lacking as most are somewhat dated, printed usually around 2005, and often mention NAS systems only briefly. Those that did go in depth into them tended to focus only on specific aspects of them.

If doing this again, I would now much rather have preferred testing the operating systems in an environment where more computers were available for load testing or providing side by side comparisons on identical hardware. Additionally, if tests had been done on campus, it is likely that better equipment could have been obtained for testing certain aspects such software RAID and network card teaming or usage of multiple switches. While I do have multiple computers that were suitable for performing many tests, the gigabit router that was used

proved to be too lacking for extensive testing as performance on it was sporadic with multiple computers transferring data through it simultaneously while the dedicated switch used for testing the final implementation server at the university was quite solid in that regard. In regards to commercial NAS devices, I do not feel that it is necessary to test them as so many websites provide reviews on them. The performance comparisons proved to be almost pointless I feel as the hardware used for testing the different operating systems seemed to be the limiting factor in many ways and this is something else that would have benefited from being done on the campus as well.

As to the final implementation of the NAS system, improvements in regards to how it is currently operating would largely be to increase the memory available to the system. As memory prices are currently quite low, expanding this would likely provide the largest benefit at the lowest cost. If the company that develops the Dataplex caching software ever update it to support Windows 8, it may be beneficial to redo the system with it installed, provided it is reliable enough to be used in a server environment. Another change to the server that may be of some benefit would be to forego use of the solid state hybrid card entirely and move on to standard hardware that would be supported by a wider range of operating systems. The problem with that though is that the other operating systems would have to be reevaluated to verify which ones are most suited for usage in the current environment.

My final thoughts on this thesis are that in doing this thesis work, it is hoped that the information detailed in it will be beneficial to others who wish to base their studies on NAS systems and prove to be informative as well. Additionally, in setting up the server for the customer, it is intended that it will benefit them in the classroom environment for which it was set up. In all, while I cannot say that I am satisfied with the results of this thesis, I do appreciate the insights it gave me into the complexities of NAS systems and network storage in general.

BIBLIOGRAPHY

Amaris, Chris, Morimoto, Rand, Handley, Pete, Ross, David E. 2012. Microsoft System Center 2012 Unleashed. USA: Pearson Education, Inc.

Apple 2012. Apple Filing Protocol Programming Guide. PDF document.
https://developer.apple.com/library/mac/documentation/Networking/Conceptual/AFP/AFP3_1.pdf. Updated 13.12.2012. Referred 20.3.2012.

Bartosh, Michael, Faas, Ryan 2005. Essential Mac OS X Panther Server Administration. Sebastopol, CA USA: O'Reilly Media, Inc.

Carpenter, Tom 2011. Microsoft Windows Server Administration Essentials. Indianapolis, Indiana USA: John Wiley & Sons, Inc.

Choubey, Manoj K., Singhal, Saurabh 2012. IT Infrastructure and Management. Noida, India: Dorling Kindersley (India) Pvt. Ltd

Collings, Terry, Wall, Kurt 2005. Red Hat Linux Networking and System Administration, Third Edition. Indianapolis, IN USA: Wiley Publishing, Inc.

Crowley, Patrick, Franklin, Mark A., Hadimioglu, Haldun, Onufryk, Peter Z. 2005. Network Processor Design Issues and Practices Volume 3. San Francisco, CA USA: Morgan Kaufmann Publishers

Dang, Alan. Angelini, Chris 2012. ARM Vs. x86: The Secret Behind Intel Atom's Efficiency. WWW-document. <http://www.tomshardware.com/reviews/atom-z2760-power-consumption-arm,3387.html>. Updated 24.12.2012. Referred 20.2.2012

DistroWatch.com 2012. rPath Linux. WWW-document.
<http://distrowatch.com/table.php?distribution=rpath>. Updated 14.8.2012. Referred 31.3.2013

Dusseault, Lisa 2004. WebDAV: next-generation collaborative Web authoring. Upper Saddle River, NJ USA: Pearson Education, Inc.

FreeBSD Handbook 2013. WWW-documents.
<http://www.freebsd.org/doc/en/books/handbook/> Updated 2013. Referred 14.4.2013

Germain, Jack M. Whither OpenSolaris? Illumos Takes Up the Mantle. WWW-document.
<http://www.linuxinsider.com/story/76669.html>. Updated 20.11.2012. Referred 10.4.2013

Heath, Steve 2003. Embedded Systems Design. Burlington, MA USA: Newnes

Heger, Dominique A. 2008. Quantifying IT Stability. Bloomington, IN USA: iUniverse

Held, Gilbert 2012. Making Your Data Center Energy Efficient. Boca Raton, FL USA: Taylor & Francis Group, LLC

Holtznider, Bill, Jaffe, Brian D. 2007. IT Manager's Handbook: Getting your new job done. San Francisco, CA USA: Morgan Kaufmann Publishers

Javvin Technologies Inc. 2005. Network Protocols Handbook. Saratoga, CA USA: Javvin Technologies Inc.

Kozierok, Charles 2005. The TCP/IP Guide. San Francisco, CA USA: No Starch Press, Inc.

Layton, Jeffery B. 2010. Cool User File Systems: GlusterFS. WWW-document. <http://www.linux-mag.com/id/7833/>. Updated 11.8.2010. Referred 21.2.2012

Lehmann, Friedrich Wilhelm 2007. Linux implementation for the ISP & data center. USA: Lulu Press

Long, James 2006. Storage Networking Protocol Fundamentals. Indianapolis, IN USA: Cisco Press

Matzan, Jem 2007. The FreeBSD 6.2 Crash Course. O'Reilly Media, Inc.

Miller, Philip M. 2009. TCP/IP - The Ultimate Protocol Guide: Volume 1 - Data Delivery and Routing. Boca Raton, Florida USA: BrownWalker Press

Nexentastor 2010. nexentastor.org Open Source Projects. WWW-documents. <http://www.nexentastor.org>. Updated 2012. Referred 10.4.2013

O'Keefe, Matthew 2012. Accelerating NAS via Hardware. WWW-document. <http://blogs.hds.com/hdsblog/2012/05/part-2-of-hitachi-nas-siliconfs-object-based-file-system.html>. Updated 30.5.2012. Referred 3.4.2013

Openfiler 2012. Openfiler commercial website. WWW-documents.
<http://www.openfiler.com>. Updated 2012. Referred 6.4.2013

Raggi, Emilio, Thomas, Keir, Parsons, Trevor, Channelle, Andy, van Vugt, Sander 2010. Beginning Ubuntu Linux, Fifth Edition. New York, NY USA: Springer Science+Business Media, LLC.

Rinnen, Pushan, Cox, Roger W., Passmore, Robert E. 2011. Magic Quadrant for Midrange and High-End NAS Solutions. PDF-document.
http://www.twintechology.co.uk/Scale/Downloads/MQ_Gartner%20Q1_2011.pdf. Updated 24.3.2011. Referred 21.2.2012

SAS 2012. SAS Acquires Key rPath Assets for Broader Deployment of SAS Solutions. WWW-document. <http://www.sas.com/news/preleases/rpath-asset-acquisition.html>. Updated 30.11.2012. Referred 31.3.2013.

Shelly, Gary B., Vermaat, Misty E. 2012. Discovering Computers Complete: Your Interactive Guide to the Digital World. Boston, MA USA: Course Technology, Cengage Learning

Shimpi, Anand Lal 2009. WWW-document <http://www.anandtech.com/show/2738/8>. Updated March 18, 2009. Referred 4.5.2013

Smith, Roderick W. 2004. The Definitive Guide to Samba 3. New York, NY USA: Apress

Stokes, Jon 2010. Intel's NAS-specific Atom platform hastens PCification. WWW-document. <http://arstechnica.com/business/2010/03/intels-nas-specific-atom-platform-hastens-pcification/>. Updated 11.3.2010. Referred 21.2.2012

Tate, Jon, Beck, Pall, Ibarra, Hugo H., Kumaravel, Shanmuganathan, Miklas, Libor 2012. Introduction to Storage Area Networks and System Networking. IBM International Technical Support Organization. PDF document.
<http://www.redbooks.ibm.com/redbooks/pdfs/sg245470.pdf>. Updated 17.11.2012. Referred 25.3.2013.

The Open Group 1997. Frontmatter. WWW-document.
<http://pubs.opengroup.org/onlinepubs/9638599/front.htm>. Updated 2.1997. Referred 11.3.2013.

Trigdell, Andrew 1998. Samba history. WWW-document.
<http://www.rxn.com/services/faq/smb/samba.history.txt>. Updated 10.1998. Referred 14.3.2013

Troppens, Ulf, Müller-Friedt, Wolfgang, Wolafka, Rainer, Erkens, Rainer, Haustein, Nils 2009. Storage Networks Explained: Basics and Application of Fibre Channel SAN, NAS, iSCSI, InfiniBand and FCoE, Second Edition. Heidelberg, Germany: dpunkt. verlag GmbH

Wallen, Jack 2010. Remote Administration with Linux. WWW-document.
<https://www.linux.com/learn/tutorials/342639:remote-administration-with-linux>. Updated 25.8.2010. Referred 10.4.2013

Waring, Becky 2007. How to Buy Network-Attached Storage Drives. WWW-document.
<http://www.pcworld.com/article/136414/article.html>. Updated 12.9.2007. Referred 20.2.2012

APPENDIX

1 Abbreviations

ACL - Access Control List

AFP - Apple Filing Protocol

API - Application Programming Interface

ASIC - Application-Specific Integrated Circuit

CIFS - Common Internet File System

CPU - Central Processing Unit

CUPS - Common Unix Printing System

DARPA - Defense Advanced Research Projects Agency

DAS - Direct Attached Storage

DVD - Digital Video Disc/Digital Versatile Disc

eSATA - external Serial AT Attachment

ESCON - Enterprise Systems Connection

ext3/ext4 - third extended file system/fourth extended file system

FCIP - Fibre Channel over Internet Protocol
FCIP - Fibre Channel over IP
FCoE - Fibre Channel over Ethernet
FCP - Fibre Channel Protocol
FICON - Fibre Connection
FPGA - Field-Programmable Gate Array
FreeBSD - Free Berkeley Software Distribution
FTP - File Transfer Protocol
GB - Gigabyte
Gbps - Gigabits per second
GUI - Graphic User Interface
HBA - Host Bus Adaptor
HDD - Hard Disk Drive
HFS+ - Hierarchical File System
HTTP - Hypertext Transfer Protocol
ID - Identification
IETF - Internet Engineering Task Force
iFCP - Internet Fibre Channel Protocol
IO - Input/Output
IP - Internet Protocol
IPFC - Internet Protocol over Fibre Channel
IPSec - Internet Protocol Security
iSCSI - internet Small Computer System Interface
ISO - International Organization for Standardization
LAN - Local Area Network
MAC - Media Access Control
MB - Megabyte
Mbps - Megabits per second
MBps - Megabytes per second
mFCP - Metro Fibre Channel Protocol
NAS - Network Attached Storage
NetBIOS - Network Basic Input/Output System
NFS - Network File System
NIC - Network Interface Card
NTFS - New Technology File System

OSI - Open Systems Interconnection

PATA - Parallel AT Attachment

PC - Personal Computer

PCI - Peripheral Component Interconnect

RAID - Redundant Array of Independent Disks

RAM - Random Access Memory

ReFS - Resilient File System

RFC - Request For Comments

rpm - revolutions per minute

RTOS - Real-Time Operating System

SAN - Storage Area Network

SATA - Serial AT Attachment

SCSI - Small Computer System Interface

SFTP - Secure File Transfer Protocol

SMB - Server Message Block

SMB/CIFS - Server Message Block/Common Internet File System

SoC - System on a Chip

SSA - Serial Storage Architecture

SSD - Solid State Drive

SSH -Secure Shell

SSHD - Solid State Hybrid Drive

TB - Terabyte

TCP - Transmission Control Protocol

TCP/IP - Transmission Control Protocol/Internet Protocol

UDP - User Datagram Protocol

UFS/UFS2 - Unix File System

USB - Universal Serial Bus

WAN - Wide Area Network

WebDAV - Web-based Distributed Authoring and Versioning

WOL - Wake-on-LAN

XFS - Extended File System

ZFS - Originally meant “Zetabyte File System” but no longer means anything